

**ENCRYPTION KEY MANAGEMENT SYSTEM
USING MULTIPLE SMART CARDS**

5

1. Technical Field:

The present invention relates to the field of encryption technology. More particularly, the present invention relates to encryption key management.

10

2. Description of Related Art:

To most people, cryptography is concerned with keeping communications private. It is the use of codes to convert data so that only a specific recipient will be able to read it using a decryption device referred to as a decryption key. Encryption is the transformation of data, through the use of codes, into a form that is as close to impossible as possible to read without the appropriate knowledge (a key). Its purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended, even those who have access to the encrypted data. Decryption is the reverse of encryption; it is the transformation of encrypted data back into an intelligible form. The persistent problem of cryptography is that the decryption key must be transmitted to the intended recipient for decryption and may be intercepted.

Encryption and decryption generally require the use of some secret information, the decryption key. For some encryption mechanisms, the same key is used for both encryption and decryption; for other mechanisms, the keys used for encryption and decryption are different.

Docket No. 99-064 MIS

In traditional cryptography, the sender and receiver of a message know and use the same secret key: the sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message.

5 This method is known as secret key or symmetric cryptography. The main challenge is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations, they must trust a courier, a phone system, or some other
10 transmission medium to prevent the disclosure of the secret key. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key. The generation, transmission, and storage of keys is called key management; all
15 cryptosystems must deal with key management issues. Because all keys in a secret-key cryptosystem must remain secret, secret-key cryptography often has difficulty providing secure key management, especially in open systems with a large number of users.

20 For an increasing number of reasons (financial, military, privacy, etc.), it is becoming highly desirable to store information in an encrypted form. However, even with all of the risks associated with failing to use encryption, most applications continue to store their
25 information in an unencrypted fashion, partially due to the problems associated with keeping the secret key a secret.

As encrypted tapes are created, an encryption key is used to secure the information. When encrypted tapes are restored, a rapid method of identifying the decryption key
30 is required. One popular method of key and data

Docket No. 99-064 MIS

association is to encrypt the key and store it within the tape itself. While the key is more secure than if it were not encrypted, the major risk with this approach is that if the key (usually 128-256 bits) were decoded, then the rest of the tape would be vulnerable. In an effort to solve the above-mentioned problems, the present invention is disclosed.

[illegible]

SUMMARY OF THE INVENTION

The present invention relates to encryption key
5 management through the use of smart card technology for
generation encryption and decryption keys used with
encryption/decryption algorithms. Two smart cards are
employed to generate, store, and lookup encryption keys and
decryption keys. Initially, an encryption smart card is
10 initialized with a card group value and a card number. For
each card number a reference number corresponding to the
largest of the keys that it has generated is also stored on
the smart card, initially the reference number is set to 0
and thereafter is incremented as each key is provided.
15 Additionally, a master key value is written to the
encryption card. A user must possess both the physical
smart card and the personal identification number
associated with the card in order to call for the
generation an encryption key. Upon receipt of a key
20 generation request, the Smart card creates a variable key
range variable by concatenating the card group value, card
number and reference number for the encryption key that is
to be generated. A hashing algorithm generates a digest
message from the master key and the variable key range
25 variable. The digest is used for an encryption key. The
encryption key and a variable key range variable are output
to the caller who requested the key. Encryption keys
generated with the encryption smart card may be used with
any encryption algorithm that uses an encryption key. The
30 variable key range variable, in and of itself, contains

Docket No. 99-064 MIS

nothing that will compromise the key. Therefore, it may be freely stored with the encrypted data.

Generation of a decryption key requires that the user possess a decryption smart card and PIN. A user calls for
5 a decryption key and includes the variable key range variable from the encrypted data. A hashing algorithm generates a digest message from the master key and the variable key range variable. The digest is used for a decryption key. The decryption key is output to the caller
10 who requested the key and is used to decrypt the information.

2025 RELEASE UNDER E.O. 14176

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figures 1A - 1C are diagrams representative of the prior art approach to decryption key management;

Figures 2A - 2C are diagrams representative of an approach to decryption key management in accordance with a preferred embodiment of the present invention;

Figures 3A and 3B are schematics that represent smart cards configurations in accordance with a preferred embodiment of the present invention;

Figure 4 is a flowchart depicting an encryption
20 process for encrypting information in accordance with a
preferred embodiment of the present invention;

Figure 5 is a flowchart depicting a decryption process for decrypting information in accordance with a preferred embodiment of the present invention; and

25 **Figures 6A and 6B** are flow charts depicting the key
generation process in accordance with a preferred
embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

5 **Figures 1A - 1C** are diagrams representative of the prior art approach to decryption key management. **Figures 1A and 1B** depict encrypted information, such as sensitive documents or secret information, stored in a binary, encrypted form on magnetic media, i.e. a reel and diskette respectively. For the purposes of the present invention, 10 the media could well be optical storage means, or even transmission media such as electromagnetic or optical propagation transmission means. Proximate to the encrypted data is a decryption key, which is normally stored or transmitted in an encrypted form.

15 A serious concern is that there may be a chance that an attacker can decipher the secret key during transmission or more probably, when the encrypted data is stored. In the case of a magnetic tape media, a tape record is shown on **Figure 1C**. Note that the decryption key is stored on 20 the media itself, but held in an encrypted form.

Subsequent to decoding, the decryption key is used to decipher any of the data files on the tape medium. Attackers may also attempt to clandestinely decipher the key and decode the remainder of the tape.

25 In the case of intercepting transmitted encrypted data, the encryption key may be transmitted sequentially with the encrypted data, albeit in its own encrypted form. Various schemes are employed for confusing attackers such as transmitting a number of encrypted keys with the 30 encrypted data, each decryption key being decrypted by

using a different decryption format. Thus, deciphering one key would not aid in deciphering other keys. The intended recipient knows in advance which is the proper decryption key and which are the decoy keys. However, an attacker can
5 always store the encrypted data, along with the all of the transmitted keys, and then simultaneously hack each of the decryption keys. Of course, this method of attack is severely limited by the attacker's hacking resources.

Figures 2A - 2C are diagrams representing an approach
10 to decryption key management in accordance with a preferred embodiment of the present invention. **Figures 2A** and **2B** depict encrypted data similar to those shown in **Figures 1A** and **1B**. However, rather than storing or transmitting a decryption key with the encrypted data, only a key range
15 variable (N) accompanies the encrypted data.

Figure 2C depicts a tape or transmission record in accordance with a preferred embodiment of the present invention. Note that the decryption key is not stored on the media, but instead generated by a smart card, such as
20 that shown in **Figure 2D**, using the variable N.

Importantly, each smart card contains a write-only master key (KM), which must correspond to a master key used to decrypt the data. The encrypted data cannot be decrypted from the information contained in the encrypted data and
25 the variable N. Even the possession of the decryption smart card itself does not guarantee that the particular decryption key needed for deciphering the encrypted data will be generated. The attacker must also have access to the smart card's personal identification number (PIN) in
30 order to generate the decryption key. Therefore, a

successful attacker must possess the encrypted data, the variable N, the decryption smart card which corresponds to the encryption smart card, and the decryption smart card's personal identification number (PIN) in order to generate the decryption key needed for deciphering encrypted data.

In accordance with a preferred embodiment of the present invention, the encryption key generation process is embodied in a Java Cardlet (an application that runs in the processing environment within a smart card). By using a smart card, the memory, algorithm, and persistent variables are inaccessible to potential attackers. **Figures 3A and 3B** are schematics representing smart cards in accordance with a preferred embodiment of the present invention. **Figure 3A** depicts the structure of encryption smart card 300. Smart card 300 utilizes conventional smart card technology, in that it comprises onboard processor 310, I/O 330, and onboard memory (or cache) 320.

Processor 310 communicates to a remote caller via smart card I/O 330 for verifying PINs and receiving requests for generating a key. In accordance with a preferred embodiment of the present invention, the encryption key and decryption key are identical, however, depending on the type of encryption, the data might be encrypted with a first key and then decrypted with a second key, which is similar to public and private key technology. In response to a request from a remote caller, processor 310 may retrieve information necessary for generating a decryption key from onboard memory 320. This information may include the current PIN, needed for verifying a PIN request, along with the applications and

applets needed for processing the PIN verification request. Additionally, encryption smart card 300 maintains four encryption variables 322:

- 5 1. A 256-bit Master Key (KM) that is set when the smart card is first initialized
2. A 3-byte key card number (C) that is set when the smart card is first initialized
3. A 5-byte reference number (n; initially set to 0) corresponding to each of the keys that it has generated (i.e., via a call to GetNextKey())
- 10 4. A 32-bit CardGroup number (G) which describes a set of cards using the same Master Key

The length of the values presented herein are representative of a preferred embodiment of the present invention but in practice may consist of any bit length. The master key is generated by an off-board application for the sole purpose of creating these cards and then destroyed. A one-time key is used for generating KM and destroyed. The KM is written to encryption smart card 300 when the smart card is initiated. KM must remain a secret because the key generation processes relies on three primary components: the range number (N), which may be found, unencrypted, in the encrypted information; the public domain hashing algorithms; and KM. Of the three, KM is the only secret component.

With respect to encryption variables 322, group number (G) is issued cards with the same KM. G can be any length, but a 4-byte value has been implemented. Card number (C) is a unique designation, which is given to each card within a given group. C can be any length, but it is currently implemented as a 3-byte value. Each encryption smart card

Docket No. 99-064 MIS

300 implements a key range variable (N), which is a concatenation of the card group (G), the individual card number (C), and the reference number (n) (of the form 0xGGGGCCnnnnn). For example, key card #1 would have the
5 range 0x123400100000 - 0x0012341FFFFFF and key card #2 would have the range 0x123400200000 - 0x1234002FFFFFF.

Once initialized, C, G, and KM cannot be changed. After the encryption key card generates each key (described in detail below), n is incremented by one. When n reaches
10 the boundary of the data type (e.g., 0xFFFFFF), encryption smart card 300 can no longer be used for key generation. Up to 4096 key generation cards may be created for a given KM and each card generates a unique set of keys.

Additionally, onboard memory 320 may contain
15 algorithms 324 needed for generating a new key. These algorithms include HMAC (Hashing based Message Authentication Codes), which is a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function,
20 e.g., SHA-1, in combination with a secret shared key. The Secure Hash Algorithm (SHA), the algorithm specified in the Secure Hash Standard (SHS, FIPS PUB 180-1: Secure Hash Standard, April 1995), was developed by NIST.

Here the HMAC-SHA-1 algorithms may be self executing
25 applications or applets, or may merely be accessed by an application in response to a new key request, GetNextKey(). With each request for a new key, a new 160-bit encryption key (KE) and the corresponding variable N are returned to the caller.

Docket No. 99-064 MIS

Decryption smart card 350 is similar in all respects to smart card 300 except for not storing the persistent variables key card number (C) and reference number (n) in onboard memory 320. The decryption key generation process is also similar to the encryption key generation process. Smart card 350 maintains two values:

1. A Master Key (KM), which is set when the smart card is first initialized. Once initialized, KM cannot be changed.
2. A 32-bit CardGroup number (G), which describes a set of cards using the same Master Key. This value is set upon card creation and cannot be changed.

In the decryption key generation process, N represents the identity of a particular encryption key. Callers pass N to the smart card when requesting a particular decryption key. With each request for a decryption key, a 160-bit decryption Key (KD) is returned to the caller. An unlimited number of decryption key cards (corresponding to every encryption key card with the same KM) may be created. Each decryption key card can generate a decryption key that maps to any encryption key (generated by encryption key cards with the same KM).

Figure 4 is a flowchart depicting an encryption process for encrypting information in accordance with a preferred embodiment of the present invention. The encryption process begins by having physical control over an encryption smart card (step 402).

For the purpose of the description of the encryption process, it is assumed that an encryption smart card issues only encryption keys while a decryption smart card issues

only decryption keys, however, a single smart card has the capability for issuing both encryption and decryption keys. Similarly, a single smart card may issue either encryption or decryption keys in a plurality of CardGroups (G). In
5 doing so, a user could employ a single smart card for encrypting and decrypting information for various individuals who are authorized to only one of the user's CardGroups. In either case, a smart card must be initialized with a master key (KM) for each CardGroup in
10 which the user intends to encrypt/decrypt information.

Returning to **Figure 4**, merely possessing the smart card is not enough. In accordance with a preferred embodiment of the present invention, the process of encrypting information relies on the smart card's inherent security
15 systems. The smart card will only respond to a user who enters an authorized personal identification number (PIN). The smart card must verify the PIN (step 404). Upon receiving the authorized PIN, the user can invoke the GetNextKey() routine for generating an encryption key (KE)
20 (step 406). GetNextKey() routine returns two values, a encryption key (a 160-bit value is used, but the bit length could be increased/decreased by modifying the key generation algorithm) and variable N (an 8-byte value is used, but could be any length). Variable N contains
25 nothing that will compromise the key and may be freely stored with the encrypted data. The only attack comes if all of the following are stolen by the same person: variable N, the encrypted data, corresponding decryption smart card, and the PIN for the decryption smart card.

The encryption and decryption keys are generated by small applications or applets running on the smart cards themselves. Keys used for encryption are generated on the encryption smart card, while keys used for decryption are generated on the decryption smart card. Exactly the same key generation algorithm is used for the encryption key generation and decryption key generation processes, i.e., the HMAC-SHA-1 process (described in detail with respect to **Figures 6A and 6B** below). The difference is in how the routines are called, either the GetNextKey() routine described hereinbefore or the GetKey() routine discussed in detail below.

Returning to step 406, the encryption smart card returns the encryption key KE (message digest) and the variable N. By using the encryption key value generated from the GetNextKey() routine as the encryption key for the encryption algorithm, information (or documents) can then be encrypted using any encryption algorithm. Of course, the encryption process is performed that converts unencrypted information to encrypted data Step 408).

The encryption smart card can issue a maximum number of keys equal to the maximum value of n (reference number). However, n is an incremental value that may be initialized at any value less than its maximum value. Therefore, the actual number of encryption keys generated by a particular card may vary from one key to the maximum value of n number of keys, depending on the initial value which n was set. Because the smart card might not have the capability to generate another encryption key, a check is then made to determine if n is at its last value (step 410). If the

Docket No. 99-064 MIS

smart card returns the last value of n for that CardGroup, the smart card cannot generate another encryption key for that CardGroup. The smart card should then be destroyed (step 412). Clearly, the physical card need not be
5 destroyed, but the master key value (KM) must be logically destroyed, at a minimum, to ensure that the spent card is not hacked for its KM.

Whether or not n is at its last value, the process continues by attaching the variable N (key range variable)
10 to the encrypted information (step 414). Contrary to the prior art, the present invention allows for the publication of the key range variable (N) without fear of the encrypted information being compromised. This is possible because the variable N is not a decryption key, but only a key
15 range variable. The key range variable is needed by the HMAC-SHA-1 algorithms for generating the exact decryption key that maps to the encryption key used for encrypting the information. Decryption keys can be generated using only the master key without using the variable N. However,
20 those decryption keys would not decode the information back to its original unencrypted form without using the variable N for selecting the correct decryption key for the encrypted information.

The key is used for encrypting information. Once the
25 information has been encrypted, the encryption key has no value and is only a security risk. The encryption key should then be destroyed (step 416). The process then ends.

Figure 5 is a flowchart depicting a decryption process
30 for decrypting information in accordance with a preferred

embodiment of the present invention. The decryption process is a similar process as that described above with respect to **Figure 4**. The decryption process begins with having physical control over the proper decryption smart card (step 502). Access to the decryption key generation capabilities of the smart card is granted when a user correctly enters a PIN, and it is verified by the smart card (step 504).

Upon receiving the authorized PIN, the user can invoke the GetKey() routine for generating a message digest used for a decryption key (KD) (step 506). As discussed above, the same key generation algorithm is used for the decryption key generation and encryption key generation processes. The difference between the two is how the key generation routines are called.

GetKey() has one input value and one output value. Variable N is retrieved from the encrypted data and input to the decryption smart card by the caller. In the present example, variable N is an 8-byte value, but could be any length as long as it is exactly the same value which was returned from the encryption smart card's GetNextKey() routine. Upon receipt of variable N, the variable is parsed and the CardGroup value is extracted from the concatenated variable N. That CardGroup value is checked against the CardGroup value stored on the smart card. This check ensures that the proper smart card can generate a decryption key that maps to the encryption key used to encrypt the information.

GetKey() outputs the decryption key (KD), which is a 160-bit value message digest in the present example, but

the bit length could be increased/decreased by modifying the key generation algorithm. The decryption key generated by the GetKey() routine is then used by a decryption algorithm corresponding to the encryption algorithm used to encrypt the information. The corresponding encryption algorithm is one which used an encryption key value generated from the GetNextKey() routine for the identical CardGroup and card number.

The information is decrypted using the decryption key KD (step 508). Afterwards the key is no longer needed and only poses a security risk, so it is deleted (step 510). This key can always be recreated following this same process.

Figures 6A and 6B are flow charts depicting the key generation process in accordance with a preferred embodiment of the present invention. Keys are generated by the applications or applets running on the smart cards. A physical smart card supports both encryption and decryption key generation. The difference between decryption key generation and encryption key generation is in how the routines are called and not in the hashing algorithms. The same hashing algorithm is used for both. The HMAC-SHA-1 key generation algorithm is used for the encryption and decryption key generation in accordance with a preferred embodiment of the present invention, but other hashing algorithms could be substituted without departing from the scope of the present invention. The hash function-based MAC (HMAC) utilized by the present invention uses the master key from the smart card in conjunction with a hash

Docket No. 99-064 MIS

function to produce a checksum mechanism for message authentication.

The key generation process begins by setting up the HMAC variables (step 602). Setup includes creating a variable called k_ipad, which is the same length as KM from the smart card and copy KM into it. Then creating a second variable called k_opad, which is also the same length as KM from the smart card and copying KM onto the k_opad variable. Next, 0x36 is Exclusively ORed (XOR) onto each byte in k_ipad while 0x5c is Exclusively ORed (XOR) onto each byte in k_opad (step 604).

Next, the SHA-1 hashing process is setup. Initially, the value of k_ipad is input to the SHA-1 hashing process (step 606). A check is then made to determine which key is being called for (step 608). If the caller has called the GetKey() routine, a decryption key is being requested by the caller. In that case the variable N is provided by the caller from the media containing the encrypted information (step 610). Variable N is added to the SHA-1 process (step 612), followed by adding the master key value (KM) from the smart card to the SHA-1 process (step 614). Finally, the k_opad variable is added to the SHA-1 hashing process (step 616), and the SHA1 hashing process creates a 160-bit message digest or key (step 618). In this case, a decryption key has been generated and returned to the caller.

Returning to step 608, if the caller has called the GetNextKey() routine, an encryption key is being requested by the caller. The process then flows to step 620 shown on **Figure 6B**. A test must be performed to determine if the

Docket No. 99-064 MIS

smart card has usable n values for generating a unique encryption key (step 620). If no n values remain, the key generation process cannot continue and ends, the smart card should be destroyed, as it has no further use.

5 If, on the other hand, another n value is available, variable N is created (step 622). Variable N consists of concatenating G, C, and n in the form of 0xGGGGCCncccc. After n has been used for creating variable N, the key generation process processed exactly as described above,
10 including adding the variable N to the SHA-1 process (step 612), followed by adding the master key value (KM) from the smart card to the SHA-1 process (step 614). The process ends with the k_opad variable being added to the SHA-1 hashing process (step 616), which culminates in the
15 creation of the 160-bit message digest or key (step 618). Finally, the value of n is incremented by one and the new n value is saved for creating another encryption key (step 624). In this case an encryption key is generated and returned to the caller.

20 The 160-bit message digest returned from the key generation algorithm is used during the encryption/decryption processes. Without the actual smart cards (which contain KM), variable N (the key reference number) does not convey any information about the keys.
25 Therefore, variable N can be used in the clear to uniquely identify the key used to encrypt the target information. In accordance with a preferred embodiment, N is stored on the tape itself to identify the decryption key to be used when the information is read. This method will uniquely

2025 RELEASE UNDER E.O. 14176

identify the keys to be used without the security risk of embedding the keys within the encrypted data.

The embodiment was chosen and described in order to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.